UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,444	03/29/2001	Junji Sakai	P/2041-58	1745

7590 05/21/2004

Steven Dickson
1177 Avenue of the americas
New York, NY 10036-2714

EXAMINER	
STEELMAN, MARY J	
ART UNIT	**PAPER NUMBER**
2122	6

DATE MAILED: 05/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/821,444	Applicant(s) SAKAI, JUNJI	
	Examiner Mary J. Steelman	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *5/14/01,5/9/01,4/30/04*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 March 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *#4, 5/9/01*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-33 are pending.

Priority

2. Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

Information Disclosure Statement

3. In reference to IDS received 05/09/2001, only the Abstracts have been considered, as the documents are not in English.

Drawings

4. The drawings are objected to because the ink is too light to read.. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

5. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description: FIG. 4, #34 & 36. A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Specification

6. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Content of Specification

(a) Title of the Invention: See 37 CFR 1.72(a) and MPEP § 606. The title of the invention should be placed at the top of the first page of the specification unless the title is provided in an application data sheet. The title of the invention should be brief but technically accurate and descriptive, preferably from two to seven words may not contain more than 500 characters.

A suggested title is "Branch Instruction Conversion to Multi-threaded Parallel Instructions"

7. The Specification and Claims are objected to because the type font is not clear. It is difficult to read the Specification and claims. Applicant is required either (1) to submit permanent copies of the identified parts or (2) to order a photocopy of the above-identified parts to be made by the U.S. Patent and Trademark Office at applicant's expense for incorporation in the file. See MPEP § 608.01.

Remove multiple references to Documents 1- 5 throughout the Specification, as Documents are not in English.

Claim Objections

8. Claims 5 and 20, page 66, line 4 and page 76, line 17, recite, "date dependence", should be –data dependence--.

Claims 12 and 16 recite "minimum value of data dependence..." , should be – minimum value of distance of data dependence...-- Add the words ' of distance'.

Claim 30 recites "a medium as claimed in claim 17...", should be –A medium as claimed in claim 28...-- Change '17' to '28'.

Claim 32, page 88, line 21, recites, "date dependence", should be –data dependence--.

Claim 33, page 89, line 10, recites, "A compiler as claimed in claim 31...", should be –A medium as claimed in claim 31--.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

10. Claims 1-33 are rejected under 35 U.S.C. 112, second paragraph:

Claims 2, 17, and 29 recite the limitation "the instruction step number". There is insufficient antecedent basis for this limitation in the claim.

Claim 27 recites the limitations "the BLOCK system" and "the DSP system". There is insufficient antecedent basis for these limitations in the claim.

Claims 6, 12, and 16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Note claim 12, page 70, line 20, "a difference greater than a predetermined value..." and line 25, "do not have a difference equal to or greater than the predetermined value (meaning the value is less than...)". In either case, the "equal to" value is not addressed, and thus unclear.

Claim 6, page 68, line 9, recites "...and so forth...". This makes the claim indefinite.

Claims 1-33 are generally narrative and indefinite, failing to conform to current U.S. practice. They appear to be a literal translation into English from a foreign document and are replete with grammatical and idiomatic errors.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,622,301 to Hirooka et al., in view of US Patent 6,588,009 to Guffens et al.

Hirooka disclosed a program conversion technique, which detects branch conversion (loop) opportunities through the use of profiling and statistical analysis. Data dependence is considered. Instructions are reordered to produce parallel code.

Per claims 1, 15, and 28, Hirooka disclosed:

Regarding register allocation, the determination of whether to convert a conditional branch to parallel code, instruction reordering...

(Col. 5, lines 39-46, "FIG. 1 is a configuration diagram showing an embodiment of a paralleling compiler...is a function of a paralleling compiler...in which compiler receives as an input a sequential execution source program and produces a parallel program.", col. 3, lines 9-13,

"...profile information, compiler static analysis information, or user indication information is obtained to produce a page allocation information to generate a parallel program..." Profiling and analysis are used to determine whether to convert to parallel code.)

Hirooka failed to discuss a control speculative mode, wherein a change having had an effect on a register set can be cancelled later and to execute the thread in a data-dependent speculative mode, details related to memory loads and details related to register allocation. However, Guffens provided more information regarding these features: Col. 1, lines 10-14, "...compiling source code by exposing parallelism, thereby enabling the generation of more efficient compiled code, allowing more parallel instructions to be created and leading to better utilization processor resources (register allocation)." Col. 5, lines 3-40, "...Extract the program operations and the information about the sequence of those operations (control speculative mode)...Classic data flow analysis...Construct the DFA Problem Graph...Create a set of questions relating to memory access ("DFA questions")...Create a symbolic execution data structure...contains data structures necessary to accumulate or hold information about all memory accesses...Route the control and data edges between operation for array variables using the answers to the DFA questions (data dependent speculative mode).

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as

provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 2, 17, and 29:

Regarding the selection of the branching destination basic block for a new thread to be executed in parallel...

(Col. 3, lines 2-5, "...profile information, compiler static analysis information...is obtained to generate...code such that a parallel program is generated..." Generated information is used. Col. 2, lines 53-65, "...parallel program generating method in which loops to be paralleled are detected and then a kernel loop is detected in the loops...code is generated...code is placed before a first execution loop of a main program...to thereby produce a parallel program...This improves data locality...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 3, 18, and 30:

Regarding the position of an instruction and "accumulating estimated execution cycle numbers of instructions"...

(Col. 3, line 39, "...page most frequently referred to by the processor...", col. 4, lines 8-13, "A profile information version data distribution control method is a method in which for each page, information of a processor which most frequently refers to the page is obtained from profile information ...thereby cause an operating system to optimally distribute data..." Frequency is a metric used to reorder code to parallel form.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 4, 19, and 31:

Regarding the execution of the target program, outputting profile information and converting a program to parallel code...

(Col. 3, lines 1-6, "...profile information...is obtained to generate a first touch control code such that a parallel program is generated...", col. 3. lines 43-47, "...profile information includes various information obtained by once executing, for example, a parallel program generated...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 5, 20, and 32:

Considering data dependence analysis calculations and probabilities, deciding whether to create a new thread or stop conversion to a parallel program.

(Col. 3, lines 2-5, "profile information, compiler static analysis information...is obtained to generate...code such that a parallel program is generated..." Generated information is used to determine whether to create new threads for a parallel program.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 6, 21, and 33:

Analysis of data dependence and conditional branching probabilities (obtained from profiling) to calculate whether to convert conditional branching to parallel code.

(Col. 3, lines 48-58, "A static analysis information version...compiler generates...static analysis information is analysis information which the compiler can automatically analyze." Profiling and analysis is performed in process to determine whether to convert to parallel code.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as

provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 7:

-Analyze source program, create intermediate program, generate code for target processor using parallel code: (Col. 5, lines 47-54, "In FIG. 1, compiler receives as an input sequential execution source program...produces and outputs parallel program for parallel execution, and generates an intermediate language...")

-Parallel conversion section: analyzing control flow and data flow: (Col. 5, lines 55-57, "Compiler includes a syntactic analysis section (analyze control flow / data flow) which reads in source program to syntactically analyze program...")

-Register allocation section: estimate register allocation prior to conversion to parallel code: (Col. 6, lines 7-9, "...data allocation information generator section to generate data allocation information of each processor using analysis...")

-Fork spot determination section: determine conditional branch portion to convert into parallel code, create thread instruction: (Col. 6, lines 23-24, "...detects all loops for execution of the paralleling...")

-Instruction reordering section: (Col. 2, lines 48-60, "...a parallel program generating method in which data is optimally distributed by the kernel loop to thereby improve data locality to increase the processing speed...loops to be paralleled are detected and then a kernel loop is detected in the loops...control code is generated and then the code is placed before a first execution loop of a main program...")

-Intermediate program outputting section: (Col. 5, line 50, "...generates an intermediate language...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 8:

Determine from register allocation trial and profile information where to convert conditional branch into parallel code: (Col. 3, lines 1-6, "...in the parallel program generating method of the present invention, it is also possible that profile information, compiler static analysis information (register allocation information), or user indication information is obtained to generate...code such that a parallel program is generated ...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 9:

Convert conditional branch Intermediate Program into parallel code, considering register allocation. Instructions maintain data dependence between threads. Reordering instructions. (Col. 2, line 48- col. 3, line 6, "...parallel program generating method...improve data locality...loops to be paralleled are detected...code is generated...to thereby produce a parallel program...profile information, compiler static analysis information...is obtained...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 10:

Assure correctness of register allocation. (See col. 3, lines 2-3, "profile information, compiler static analysis information...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 11:

Syntax analysis of source program to convert intermediate program to parallel code including:

-Register allocation trial: (Col. 3, lines 2-3: Static analysis and profiling are done.)

-Calculating distance of data dependence of Intermediate program. (Col. 2, lines 49-50 and 65, "improve data locality..." An intermediate language is generated (col. 5, line 50) and syntactic analysis is considered.)

-Considering the distance of data dependence, replace condition branch instructions with thread creation instructions. (The invention creates parallel code (creates threads for multiple processors) when loops are detected. Static analysis and profiling are done.)

-Reorder instructions: (Col. 2, lines 56- 60, "...code is generated...thereby produce a parallel program. (sequential instructions are reordered to create parallel code)")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 12 and 16:

Calculate a value of distance of data dependence, using value determine whether to select code for conversion to parallel code: (Col. 2, lines 50 and 65, "...improve data locality..." Distance is considered. Static and profiling of code is done.)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as

provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 13:

-Using profile information, calculate conditional branching probabilities and data dependence occurrence frequency... (Col. 3, lines 1-6, Profile and static analysis is used. Frequency is considered (col. 3, lines 35-39). Branching is converted to parallel code (col. 5, lines 60-62), "inserts the code into intermediate language to convert parallel processing thereof by a plurality of processors, and a code generator section which generates and outputs a parallel program using intermediate language converted.")

-Consider distance of data dependency, conditional branch probability, data dependency occurrence frequency, and the number of spots of different memory addresses to determine a fork destination. Replace conditional branch with thread creation instruction. (Col. 3, lines 1-6, "profile information, compiler static analysis information...is obtained to generate...code such that a parallel program is generated..." Data dependency, branch probabilities, frequency and memory addresses to determine a fork destination are analyzed. Loop instructions (conditional branch) are converted to parallel code (thread creation).)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 14:

See limitations addressed in claim 1 above.

Per claims 22 and 26:

-If it is determined that a conditional branch is a return branch of a loop, determine direction of return loop, and select branch as a fork spot. (Abstract: lines 6-11, "...a kernel loop having a longest sequential execution time is detected in the source program, Next, a data access pattern equal to that of the kernel loop is reproduced to generate a control code...The first touch control code generated is inserted in the parallel program.", col. 2, lines 54-55, "...loops (return branch) to be paralleled are detected and then a kernel loop is detected in the loops.")

-If it is not a loop, calculate a value related to distance of data dependence for the branches, use calculations to determine fork spot. (Abstract: lines 6-11, "...a kernel loop having a longest sequential execution time is detected in the source program, Next, a data access pattern (value related to distance of data dependence) equal to that of the kernel loop is reproduced to generate a control code...The first touch control code generated is inserted in the parallel program." Profile and static analysis information is obtained (col. 3, lines 2-3).)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 23:

Distance of data dependence is number of steps in the Intermediate program-distance from top of basic program. (Col. 3, lines 2-4, "...profile information, compiler static analysis information...is obtained...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claim 24:

Number of cycles estimated to be required at execution. (Col. 3, lines 2-4, "...profile information, compiler static analysis information...is obtained...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Per claims 25 and 27:

-Instruction reordering, register allocation, replace branch instruction with FORK instruction, relocate branch conditional instruction, according to whether branch condition is satisfied, decide which thread to follow, move instruction, allocate registers. (Col. 2, line 47-col. 3, line 6, "...parallel program generating method...improve data locality...loops to be paralleled are detected...code is generated...to thereby produce a parallel program...profile information, compiler static analysis information...is obtained...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Hirooka's invention to include more details regarding control and data dependent speculation, memory accesses and resource (register) optimization, as provided by Guffens', because these details are well known and contribute to efficient compiled parallel code.

Conclusion

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306 for regular communications and for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

05/12/2004

TUAN DAM
SUPERVISORY PATENT EXAMINER